Attributes and VOs: Extending the UNICORE authorisation capabilities

Arash Faroughi[1], Roozbeh Faroughi[1], Philipp Wieder[2], and Wolfgang Ziegler[1]

[1] Fraunhofer Institute SCAI, Department of Bioinformatics,
53754 Sankt Augustin, Germany
{arash.faroughi, roozbeh.faroughi, wolfgang.ziegler}@scai.fraunhofer.de
[2] Central Institute for Applied Mathematics, Research Centre Jülich,
52425 Jülich, Germany
ph.wieder@fz-juelich.de

Abstract. Reliable authentication and authorisation are crucial for both service providers and their customers, where the former want to protect their resources from unauthorised access and fraudulent use while their customers want to be sure unauthorised access to their data is prevented. In Grid environments Virtual Organisations (VO) have been adopted as a means to organise and control access to resources and data based on roles that are assigned to users. Moreover, attribute based authorisation has emerged providing a decentralised approach with better scalability. Up to now UNICORE authentication and authorisation is based on X.509 certificates only. In this paper we will present two approaches to integrate both role or attribute based authorisation using VOMS and attribute based authorisation using Shibboleth into UNICORE.

1 Introduction

In collaborative distributed environments like Grids where services are offered by multiple service providers and consumed by a large number of clients, the concept of linking authorisation to roles and attributes may help lowering the complexity of user management and user authorisation while making the process more transparent. In Grid environments the concept of Virtual Organisations (VOs) [8] are utilised as a powerful instrument for creating dynamic organisations whose members are sharing a common goal, e.g. researchers working together on a dedicated subject for a limited time using shared resources. Based on roles defined for VO members the service provider will enforce the access control for resources and data. A well known example are the four major experiments around the Large Hadron Collider [10] where scientists are organised in four corresponding VOs to share their data and get access to the computational and storage resources of EGEE [4]. In general, the VO-based approaches are relying on users being furnished with X.509 certificates from a trusted Certificate Authority (CA) operated e.g. by their home institution or on a project level. However, setting up such trusted Public Key Infrastructures (PKI) is often beyond the capabilities of

smaller institutions or regarded as imposing too much effort for large institutions with a huge number of certificates to be managed.

In collaborative environments without a PKI the concept of identity federation has been identified as a mechanism for authentication and authorisation. The most prominent system is Shibboleth [19]. Shibboleth implements a distributed approach where the user authenticates vis--vis his home institution when trying to access resources of a service provider and in turn the service provider gets access to selected attributes of this user, which are maintained by his home institution. Based on these attributes the service provider then decides on the authorisation of the user to access the requested resources.

For the rest of this paper we will use the term role-based authorisation for authorisation based on the information about a user within a VO maintained by a VO Management system and usually included in his X.509 certificate like e.g. when using the Virtual Organisation Membership Service (VOMS) [21]. In contrast, we will use the term attribute-based authorisation when attributes of a user stored at his home institution are used for taking authorisation decisions, i.e. when using Shibboleth mechanisms to retrieve these attributes.

The authentication and authorisation in UNICORE is based on plain X.509 certificates until now, which implies several limitations, e.g. single sign on (SSO) for Grid resources that are not part of a UNICORE environment is not supported as UNICORE so far only allows static explicit trust delegation (ETD) and users without a certificate could not access UNICORE resources of a UNICORE based Grid at all. Meanwhile, some of the issues were smoothed out by the GRIP project [11] and trust delegation through proxy-certificates is supported by UNICORE. However, VOs and authorisation based on information about the role of users within VOs are missing concepts in UNICORE today. UNICORE lacks also support for using attributes of a user retrieved from his home institution. Overcoming these limitations is part of the IVOM project [9] funded by the German D-Grid Initiative [3] and we will present work done in IVOM in this paper.

The remainder of the paper is organised as follows: Section 2 describes current attribute-based authorisation work. Section 3 gives a brief overview on the two basic technologies for role-based and attribute based authorisation we consider for the integration with UNICORE. Section 4 introduces the current UNICORE mechanisms for authentication and authorisation. The architectures for the integration with VOMS and Shibboleth are presented in Section 5. Section 6 describes future work and concludes the paper.

2 Related work

The Swiss National Research Network (NREN) SWITCH is setting up a Shibboleth based authentication and authorisation infrastructure (AAI) for the Swiss Research infrastructure. SWITCH has set up a federation of Swiss Identity Providers (IdP) and Service Providers (SP) and operates the necessary Short Lived Credential Service (SLCS) [15]. SWITCH also developed the VOMS At-

tribute from Shibboleth (VASH) [7] service that allows transferring the Shibboleth user attributes into VOMS. At the time of the VASH development GridShib was not considered because of its limitation to the pull-model for retrieving the attributes from the IdP. The German NREN DFN also has almost completed the setup of a federation of German IdPs and SPs [2]. DFN is finalising the accreditation of its SLCS by the EUGridPMA and is expected to go into productive operation in near future. More work on integration of the different existing VO-Management technologies and the introduction of attribute-based authorisation in D-Grid is done in the IVOM project [9]. Finally the OMII-Europe project [12] is working on integration of SAML assertions into UNICORE coming from a modified VOMS based on the emerging OGSA AuthZ standard.

3 Role-based and attribute-based Authorisation

3.1 Virtual Organization Membership Service (VOMS)

VOMS has been developed as part of the joint efforts of the European DataGrid and DataTAG projects. It is a system, which classifies users that are participating in a VO based on a set of attributes that will be granted to them upon request. These attributes will be included into Globus-compatible proxy-certificates for supporting SSO in grid-environments. VOMS consists of four main components [1]:

User Server: receives requests from a client and returning information about the user

User Client: contacts the User Server with the user's certificate, authenticates the user to the server and creates a proxy certificate with VO Fully Qualified Attribute Name (FQAN) extensions.

Administration Client: used by the VO Administrator to add/delete and change VO-Attributes like roles and groups.

Administration Server: accepts requests from client to update the database.

Prior to getting access to the Grid the user must execute the voms-proxy-init to generate a proxy-certificate, similar to the grid-proxy-init command of the Globus Toolkit. The main difference is that VOMS includes the authorisation information of the user into the proxy certificate retrieved from the VOMS-Server [6] resulting in an Attribute Certificate (RFC 3281). List 1 describes the procedure:

1. Using certificates the user and VOMS Server authenticate each other;
2. The user sends a request to the VOMS Server, which is signed by it;
3. The VOMS Server verifies the user's identity and checks the syntax of the request;
4. The VOMS Server sends the required authorisation information as an attribute certificate back to the user;
5. The user validates the response of the server;
6. Optionally, the user repeats this process for other VOMSes;
7. The user creates the proxy certificate and inserts the received authorisation information into a (non-critical) extension of the proxy-certificate.
8. The user may add user-supplied authentication information (e.g. Kerberos tickets)

List 1: VOMS operations

3.2 Shibboleth

The open source system Shibboleth supports an Attribute Based Access Control model. Shibboleth is a federated identity management system, developed by Internet2 and supports authorisation decisions based on the attributes of the users. It uses the Security Assertion Markup Language (SAML) to implement SSO across or within organisational boundaries. The three major components of Shibboleth are [13]]:

The **Identity Provider** (IdP) is responsible for asserting authentication and authorisation information about their Shibboleth-user. The IdP is located at the home organisation of a user. The IdP has two major services, the Handle Service (HS) and the Attribute Authority (AA). The HS authenticates the user and issues an Attribute Query Handle in the form of a signed SAML response. The AA responds to requests for user-attributes by the Attribute Requester (AR) of the Service Provider.

The **Service Provider** (SP) is offering protected resources to customers. The SP decides and enforces the authorisation to access resources. The SP consumes SAML Assertions. The most important components of the SP are the Assertion Consumer Service (ACS), the Attribute Requester (AR) and the Resource Manager (RM). The ACS validates Authentication assertions from the HS of an Identity Provider. The AR is responsible to request attributes from the user's IdP. The RM makes authorisation decisions based on the user's attributes.

The **Where Are You From Service** (WAYF) may be used to establish the communication between the Service Provider and the Identity Provider of the user. It provides a mechanism for routing users from a resource to their point of login. The WAYF Service shows a list of IdPs where users have to select their IdP.

List 2 described the process of the authentication and the transfer of the user attributes to the SP.

1. A user tries to access a resource, which is located at the Service Provider
2. The Service Provider needs to identify the home organisation where the user is known. Therefore the Service Provider redirects the user to the WAYF Service.
3. The WAYF service shows a list of Identity Providers.
4. The user selects his Identity Provider aka home organisation.
5. The WAYF service redirects the user to his Identity Provider.
6. His home organisation asks the user to provide his/her login credentials.
7. The user provides his/her credentials to the home organisation.
8. After a successful AuthN the Identity Provider creates a handle and forwards this to the Service Provider.
9. The Service Provider sends an attribute request to the Identity Provider of the user by sending the received handle.
10. The Attribute Authority (AA) verifies the Handle. After a successful validation the AA follows the rules of the Attribute Release Policy when deciding whether or not to release an attribute. The AA sends the released attributes to the Service Provider. The Resource Manager makes authorisation decisions based on the user's attributes.

List 2: Ten steps accessing protected resources using Shibboleth [20] [13]:

4 UNICORE authentication and authorisation model

The target sites comprising a UNICORE Grid need to verify the identity and access rights of users who want to execute tasks on the target sites, and they must, in addition, verify that the tasks they receive for execution belong to the appropriate users. To achieve this, the UNICORE security model specifies the usage of permanent X.509 certificates (which are issued by a Certification Authority (CA)) to authenticate and authorise users, to authenticate UNICORE server components, and to sign jobs and software plugins [18]. In the course of this section we contemplate user authentication and user authorisation since both mechanisms are fundamental to the enhancements we present later. The user's X.509 certificate is used to provide single-sign-on in the UNICORE client. The client unlocks the user's keystore once the correct password is entered at start-up, which implies that no further password requests are demanded from the user. To authenticate the user, the client has to present the user's X.509 certificate to the UNICORE Gateway. The certificate is issued by a CA that is being trusted within a UNICORE Grid. This implies that the signer certificate of this particular CA is included in the server components of UNICORE. Please note that UNICORE neither prescribes a specific Certification Authority nor is limited to the usage of only a single CA.

To authorise users, certificates are mapped to local accounts (i.e. in general a standard UNIX uid/gid), which may be different at each site due to existing naming conventions. The rules that describe the mapping of a user's abstract identity (which is contained in the certificate) to the concrete one at the local site are contained in the UNICORE User Data Base (UUDB). Through the UUDB sites retain full control over the authorisation of users and over the underlying rules leading to the acceptance or rejection of a particular individual based on the distinguished name or other information that might be contained in the certificate. UNICORE can handle multiple user certificates (abstract identities) of a single user, i.e. it permits a client to be part of multiple, disjoint Virtual Organisations. In addition the client offers the possibility to configure different project accounts so to allow users to select different accounts for different projects on one execution system or to maintain different roles with different privileges.

The private key contained in the client keystore is also used to sign each UNICORE job (and all the nested sub-jobs). This mechanism protects against tampering while the job is transmitted over insecure connections and it allows to verify the identity of the owner at the receiving server without it trusting the intermediate sites which forwarded the job.

UNICORE, apart from extensions implemented to realise interoperation with other Grid middleware like Globus, does natively not support proxy certificates. Instead it supports an explicit trust delegation [14] that allows trusted "agents" in the Grid to create jobs on behalf of end-users. This mechanism allows services like brokers, schedulers, or third-party SLA negotiators to be integrated into a UNICORE Grid without breaching UNICORE's security model.

5 UNICORE integration with VOMS and Shibboleth

Goals: The goals of the UNICORE integration with VOMS and Shibboleth are to extend UNICORE with attribute and role based authorisation, though to keep the modification of the UNICORE Client as minimal as possible and to keep the UNICORE authentication mechanisms as much as possible.

5.1 The Integrated Architecture of VOMS and UNICORE

To support VO-based Authorisation in UNICORE, a VO-Module is needed, which creates the user certificates with VO-authorisation information. VOMS creates credentials in form of proxy certificates. For the integration of VOMS and UNICORE a VOMS-plugin and an extension of the UUDB will be implemented:

The **VOMS-Plugin** is the VO-module for UNICORE and allows the user to specify his request for creating proxy certificates including the vo/group/role-information. The VOMS-Plugin generates a proxy certificate with VOMS-specific extensions for the end-user (by using the VOMS-command voms-proxy-init and attaches it to the Abstract Job Object (AJO) encapsulated as a site-specific security object (SSO-Object) [5].

Extended UUDB: UNICORE maps with the UUDB the identity of the end-user (DN-subject of the end-user certificate) to a local account. To support Role-Based Access Control (RBAC) or Attribute Based Access Control (ABAC) authorisation, the UUDB authorisation mechanism must be extended. For the UNICORE-VOMS integration, a new UUDB implementation is needed, which does the mapping on basis of VOMS-FQANs.

UNICORE-VOMS-Architecture: The VOMS plugin is integrated into the UNICORE client and doesn't modify the authentication mechanism of UNICORE. It extends the UNICORE client with the functionality described above. The UUDB implementation of UNICORE 5 must be replaced by the VOMS-extended UUDB. UNICORE provides a simple way for inserting plugins and changing the UUDB-Implementation. Therefore, UNICORE can be extend with the VOMS-Implementation using standard features.

List 3 and Figure 1 illustrate the architecture of the UNICORE/VOMS integration and the communication-procedure:

1. The user authenticates to the UNICORE client using his permanent user certificate;
2. The user specifies his VOMS-request by using the VOMS-Plugin of the UNICORE client;
3. The VOMS-Plugin checks the syntax of the request, validates the users identity and sends the request to the VOMS Server:
4. The VOMS Server validates the request and sends the required authorisation information as an attribute certificate to the VOMS-Plugin;
5. The VOMS-Plugin generates a proxy-certificate with the received VOMS extensions.
6. The user submits a job with the usual UNICORE mechanism. The VOMS-Plugin encapsulates the proxy certificate in the AJO SSO. The UNICORE Client sends the AJO via the UNICORE Protocol through the Gateway and the NJS component.
7. The VOMS-extended UUDB maps the VO-FQAN of the user's proxy certificate to a local account.

List 3: Interaction of UNICORE and VOMS

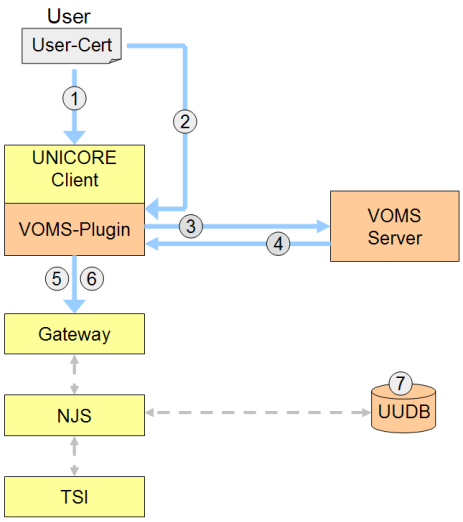

Fig. 1. Architecture of the UNICORE and VOMS integration

5.2 UNICORE Integration with Shibboleth

The major modification of UNICORE with respect to the Shibboleth integration is the UUDB, therefore the effort to integrate Shibboleth with higher UNICORE versions are expected to be small. The main tasks to extend UNICORE with shibboleth-based authorisation are:

UNICORE Authentication with a Short Lived Credential (SLC): The exchange in Shibboleth is based on assertions between an Identity Provider and a Service Provider. The AAI in UNICORE relies on the usage of X.509 certificates. With MyProxy a Shibboleth Identity can be translated to Grid Identity by generating a SLC. Using SLC the authentication mechanisms in UNICORE are largely unchanged. MyProxy issues a short lived X.509 Credentials after a successful user authentication at a Shibboleth Identity Provider. The short lived Credentials have a maximum Lifetime of 1 million seconds. To use SLC for the UNICORE user-authentication, the UNICORE Client, the UNICORE Gateway and the UNICORE NJS must trust the MyProxy Certification authority.

Attribute-Based UUDB: UNICORE maps the identity of the end-user (DN-subject of the end-user certificate) to a local account using the information stored in the UUDB. To support ABAC-Authorisation, the UUDB authorisation mechanism must be extended. For the UNICORE-Shibboleth integration, an extended UUDB implementation is needed, which does the mapping to the permissions based on the attributes.

5.3 Necessary Changes of the Shibboleth-Framework:

Shibboleth's HTTP-Redirects: The usual web-based Shibboleth mechanisms of redirecting the user accessing the service provider to the WAYF server and the IdP is not practical in Grid environments submitting unattended batch jobs [17]. For this reason, the UNICORE/Shibboleth-approach does the user-authentication before requesting a protected resource. This will be realised by a module collecting the user's authentication and authorisation information for creating SLCs and convey those to the UNICORE System. The authentication and authorisation will be done in UNICORE with these SLCs.

The **WAYF Service** has a list of known and trusted Identity Providers in a Shibboleth Federation. The user first interacts directly with the WAYF service and then tries to access a resource.

UNICORE-Shibboleth-Architecture: An external Shibboleth-UNICORE module will be implemented for the creation of SLCs. The authentication is done by the IdP and the SLC is issued by MyProxy (see List 4 and Figure 2).

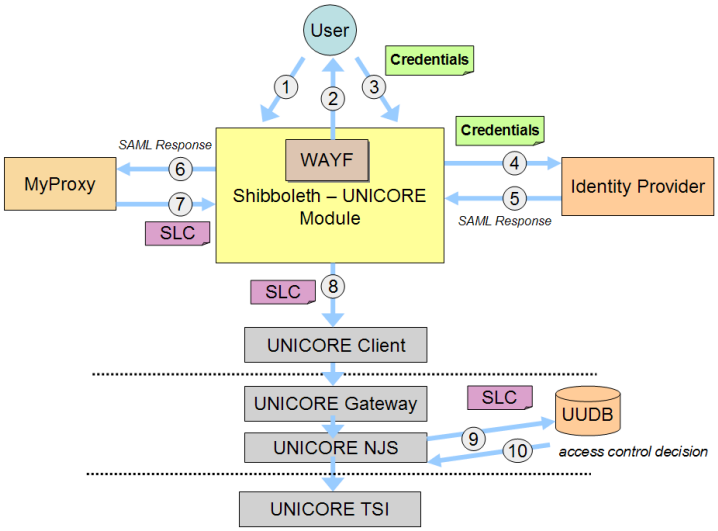

Fig. 2. Integration of UNICORE and Shibboleth

1. The user runs the Shibboleth-UNICORE Module
2. The Module fetches an up-to-date list of IdPs from the WAYF server, presents the list of Identity Providers and makes a Shib-Authentication Request.
3. The user selects his/her identity provider and gives his/her Credentials.

4. The Shibboleth-UNICORE Module authenticates the user to the Identity Provider.
5. After a successful authentication the IdP returns a SAML Response to the Shibboleth-UNICORE Module. The Response contains an authentication assertion and an attribute assertion.
6. The Shibboleth-UNICORE Module presents the Response to MyProxy.
7. MyProxy generates an X.509 credential, inserts the attribute assertion into the certificate, and returns the credential to the Shibboleth-UNICORE Module.
8. The Module runs the UNICORE Client and uses the SLC for the UNICORE user authentication.
9. The NJS sends the SLC to the UUDB.
10. Based on the attributes of the SLC the UUDB response with an access decision.

List 4: Interaction of UNICORE and Shibboleth

6 Future work

Within the IVOM project the feasibility studies and the design of the two architectures have been completed recently and we started with the implementation of the UNICORE extensions. In the meantime the IVOM project will also provide a version of GridShib for the D-Grid which will then be integrated also allowing GridShib users accessing UNICORE resources. We also plan to integrate the VO Membership Registration Services [16] database of VO attributes with UNICORE in the next version. Also, because in D-Grid the currently used UNICORE 5 will be replaced by UNICORE 6 in 2008 we plan to integrate UNICORE 6 when it will become available. Once available we will switch to the D-Grid Short Lived Credential Service (SLCS) operated by the German DFN for the creation of SLCs. Finally, we will co-operate with the OMII-Europe project [12], which is working on similar extensions of UNICORE.

7 Acknowledgements

Some of the work reported in this paper is funded by the German Federal Ministry of Education and Research through the D-Grid project under grant #01AK800A. This paper also includes work carried out jointly within the CoreGRID Network of Excellence funded by the European Commission's IST programme under grant #004265.

References

1. R. Alfieria, R. Cecchinib, V. Ciaschinic, and L. dellAgnello. From gridmap-file to voms: managing authorization in a grid environment. *Future Generation Computer Systems*, 21 (4):549 – 558, 2005. http://www.fis.unipr.it/lca/grid/doc/from-gridmap.pdf.
2. DFN-AAI: Authentication and Authorisation Infrastructure in DFN. Website: https://www.aai.dfn.de/der-dienst.html, (in German), last visited: June 15, 2007.

3. D-grid initiative. Website: http://www.d-grid.de/index.php?id=1&L=1, last visited: June 15, 2007.
4. EGEE - Enabling Grids for E-sciencE. Website: http://www.eu-egee.org/, last visited: June 15, 2007.
5. Philipp Wieder et al. Grid interoperability project. Technical report, FZJ Jülich Germany, 2002.
6. R. Alfieri et al. From gridmap-file to VOMS - managing authorization in a Grid environment. Technical report, INFN Parma and University of Parma, 2004. http://grid-auth.info.it/docs/voms-FGCS.pdf.
7. P. Flury, V. Tschopp, T. Lenggenhager, and C. Witzig. Shibboleth Interoperability with Attribute Retrieval through VOMS. Technical report, EGEE, 2006. https://edms.cern.ch/file/807849/1/EGEE-II-MJRA1.5-807849-v0.95.pdf.
8. I. Foster, C. Kesselman, and S.Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *Journal of High Performance Computing Applications*, 15 (3):200 – 222, 2001. www.globus.org/research/papers/anatomy.pdf.
9. Interoperability und integration of vo-management technologies in d-grid. Website: http://www.d-grid.de/index.php?id=314&L=1, last visited: June 15, 2007.
10. LHC - The Large Hadron Collider. Website: http://lhc.web.cern.ch/lhc/, last visited: June 15, 2007.
11. D. A. Nicole. UNICORE and GRIP: Experiences of Grid Middleware Development. In *Proceedings of 2005 International Conference on Grid Computing and Applications*, pages 11 – 17. ECS, June 2005. Online at: http://eprints.ecs.soton.ac.uk/11889/01/gca_final.pdf.
12. Open Middleware Infrastructure Institute Europe - OMII-Europe. Website: http://omii-europe.org/OMII-Europe/, last visited: June 15, 2007.
13. Tom Scavo and Scott Cantor. Shibboleth architecture, technical overview. Technical report, 2005. http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf.
14. D. Snelling, S. van den Berghe, and V. Li. Explicit trust delegation: Security for dynamic grids. Technical report, December 2004. FUJITSU Scientific and Technical Journal, 40(2):282.294.
15. Short Lived Credential Service (SLCS). Website: http://www.switch.ch/grid/slcs/, last visited: June 15, 2007.
16. VO Membership Registration Service. Website: http://www.uscms.org/SoftwareComputing/Grid/VO/, last visited: June 15, 2007.
17. V. Welch, T. Barton, K. Keahey, and F. Siebenlist. Attributes, anonymity, and access - shibboleth and globus integration to facilitate grid collaboration, 2005. Online: http://grid.ncsa.uiuc.edu/papers/gridshib-pki05-final.pdf.
18. Ph. Wieder, T. Goss-Walter, R. Letz, T. Kentemich, and H.-C. Hoppe. An analysis of the unicore security model. Technical report, Global Grid Forum. Grid Forum Document - Informational 18 (GFD-I 18).
19. Shibboleth. Website. Online: http://shibboleth.internet2.edu/.
20. The Swiss Education and Research Network. Website. Online: http://www.switch.ch/aai/demo/medium.html.
21. Virtual Data Toolkit: VOMS-Documentation. Website. Online: http://vdt.cs.wisc.edu/VOMS-documentation.html.